



Выборка данных

Команда SELECT – выборка данных

Общий синтаксис:

```
SELECT [{ ALL | DISTINCT }] { список_вывода | * }  
  FROM имя_таблицы1 [ алиас1 ] [, имя_таблицы2 [ алиас2 ],...]  
  [ WHERE      условие_отбора_записей ]  
  [ GROUP BY { имя_поля | выражение },... ]  
  [ HAVING      условие_отбора_групп ]  
  [ UNION [ALL] SELECT ...]  
  [ ORDER BY имя_поля1 | целое [ ASC | DESC ]  
    [, имя_поля2 | целое [ ASC | DESC ],...]];
```

Примеры:

```
select * from departs;
```

```
select name, post from emp;
```

Однотабличные запросы

```
SELECT  col1, col2, col3  
         FROM  table_name  
         WHERE expression
```

Выборка из таблицы

```
SELECT ID, name, countrycode, pulation  
FROM city;
```

Арифметика в столбцах

```
SELECT id*200, name, countrycode+10  
  FROM city;
```

Синонимы для столбцов

```
SELECT    2+3 sign,  countrycode cc  
FROM      city;
```

Предикат условия where

```
SELECT * FROM city WHERE id = 123;
```

#конструкция для набора IN

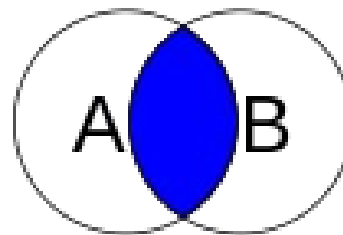
```
SELECT *  
  FROM city  
 WHERE id in (123, 7, 5);
```

#выбор по шаблону LIKE

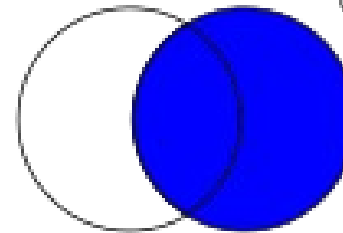
```
SELECT * FROM city  
 WHERE name like '%Petersburg%';
```

Многотабличные запросы

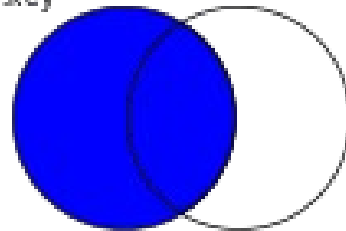
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key



SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key

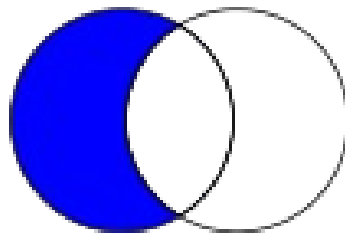


SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key

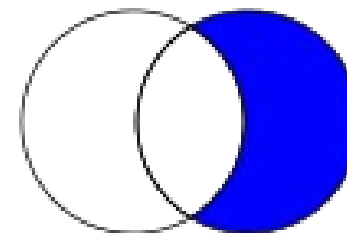


SQL JOINS

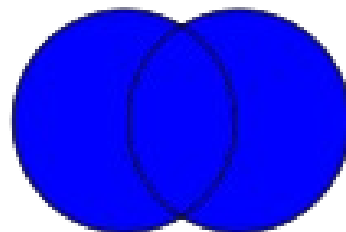
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL



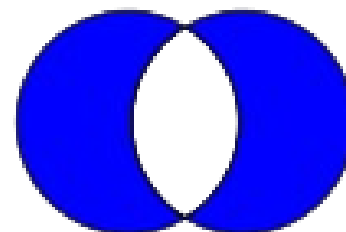
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL



SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key



SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL



Многотабличные запросы

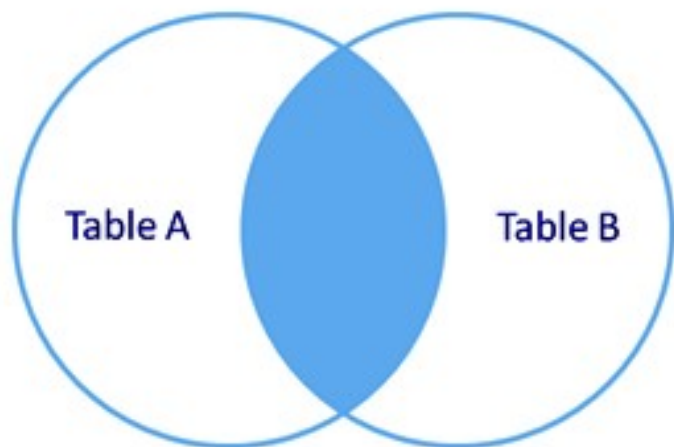
CROSS JOIN: Перекрестное соединение, возвращает комбинации каждой записи первой таблицы с каждой записью второй таблицы.

INNER JOIN: Внутренним соединением называется перекрестное соединение, из результатов которого часть записей исключена по условию запроса.

LEFT JOIN: В левом внешнем соединении для КАЖДОЙ ЗАПИСИ ЛЕВОЙ таблицы ищется соответствие среди записей правой таблицы.

RIGHT JOIN: Правое внешнее соединение ищет в левой таблице соответствия для правой таблицы.

INNER JOIN



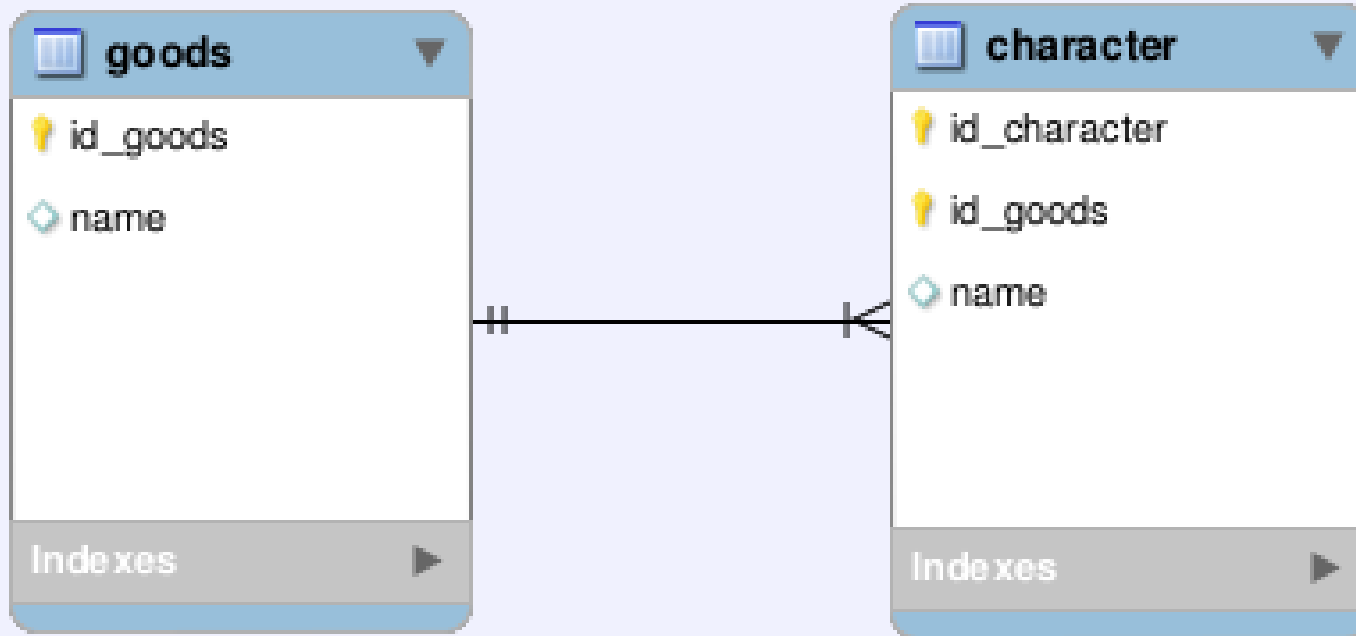
pic. Inner join

Шаблон запроса:

```
SELECT    a.name , b.value
FROM      table1 a, table2 b
WHERE     a.id = b.id
```

Модель

Конструктор товаров



Данные

GOODS

ID_GOODS	NAME
1	Книга
2	Жесткий диск
3	Системный блок
4	Монитор

CHARACTER

ID_CHARACTER	ID_GOODS	NAME
1	1	Автор
2	1	Кол-во страниц
3	1	Издательство
4	1	Год выпуска
5	2	Объем
6	2	Скорость вращения шпинделя
7	2	Интерфейс
8	3	Жесткий диск
9	4	Процессор

Запрос

(связь по ключу PK = FK)

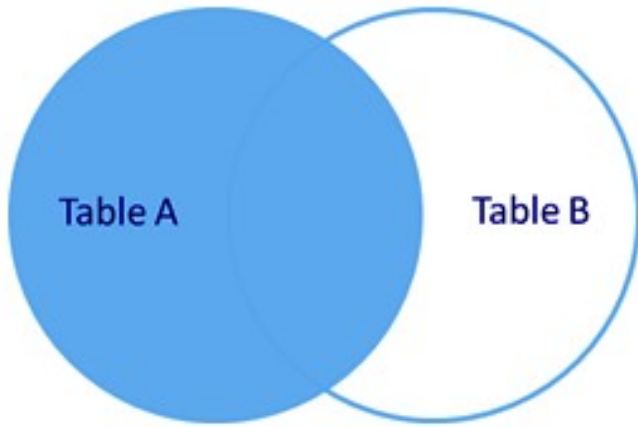
```
SELECT g.name, ch.name  
  FROM goods g, character ch  
 WHERE g.id_goods = ch.id_goods  
       AND g.id_goods = 1
```

Какой результат получим ?

CROSS JOIN

```
SELECT g.name, ch.name  
      FROM goods g, character ch
```

LEFT JOIN



pic. Left join

получим все товары у
которых заданы и не заданы
характеристики.

```
SELECT a.name, b.name  
FROM goods a left join characters b  
on a.id_goods = b.id_goods;
```


фильтр is not null

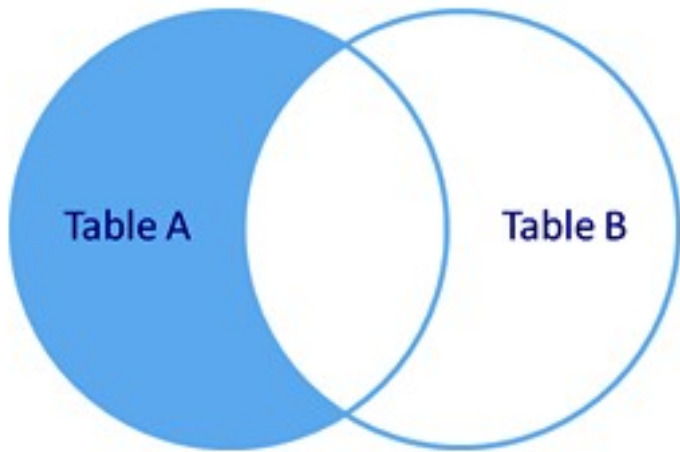
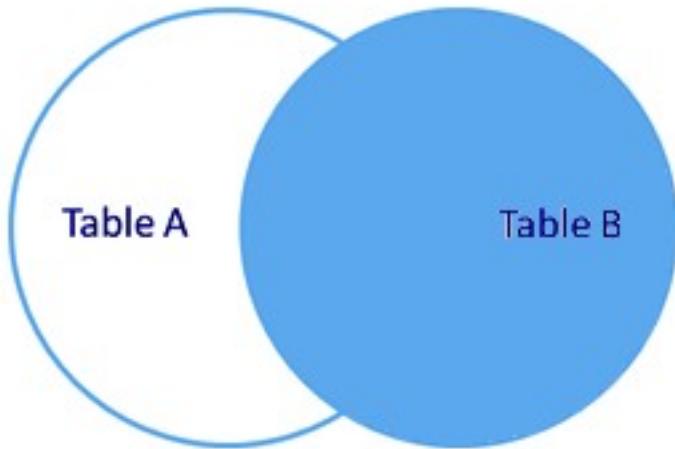


рис. Left outer join с фильтрацией по полю

Получим только те товары у которых не задана характеристика.

```
SELECT a.name, b.name
FROM goods a
      left join characters b
      on a.id_goods = b.id_goods
WHERE b.name IS NULL ;
```

RIGHT JOIN



получим все характеристики
у которых заданы и не
заданы товары.

```
SELECT a.name, b.name
FROM goods a
      right join characters b
on a.id_goods = b.id_goods;
```

Полное объединение

Союзы

```
SELECT n from numders1;  
UNION  
SELECT n from numbers2;
```

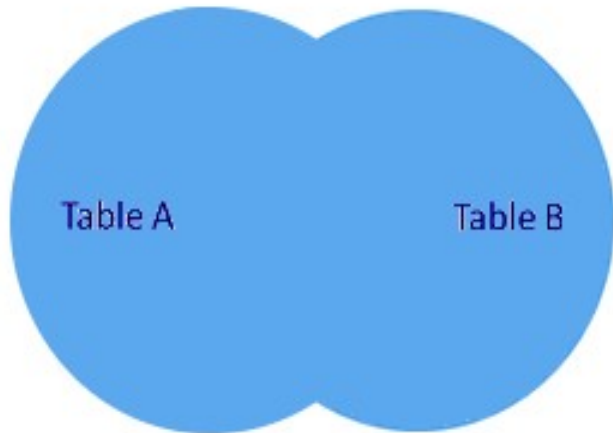
UNION — Объединяет в одну таблицу результаты 2-х и более запросов.

UNION ALL — Для получения списка со всеми дубликатами.

INTERSECT — Возвращает пересечение результатов нескольких запросов.

EXCEPT — Возвращает исключение результатов второго запроса из первого.

FULL OUTER JOIN



получим полное
пересечение
соединений

```
SELECT a.name, b.name
FROM goods a
left join characters b
ON a.id_goods=b.id_goods
UNION
SELECT a.name, b.name
FROM goods a
right join characters b
ON a.id_goods=b.id_goods;
```

Задание.

Для разрабатываемой модели построить
многотабличные запросы исходя из
бизнес-логики.